Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (currently amended) A computer-implemented method for managing a re-usable resource comprising:

dividing a pool of integers into groups that include unique sets of integers; and

initializing, in computer memory, a doubly linked list that represents one of said groups of integers in response to a request for a free integer, wherein said one group of integers includes said free integer.

2.   (currently amended) The computer-implemented method of claim 1 further including deleting, from said computer memory, an active doubly linked list that represents one of said groups of integers when all of the integers in said one group are free.

3.   (currently amended) The computer-implemented method of claim 1 wherein initializing a doubly linked list includes:

establishing an array of linked list elements that defines said group of integers, wherein each linked list element includes a next pointer for identifying a next linked list element and a previous pointer for identifying a previous linked list element;

forming a doubly linked list, from said linked list elements, that includes all of the integers in said group of integers; and

establishing a head element having a next pointer for identifying one end of said doubly linked list and a previous pointer for identifying the other end of said doubly linked list.

4.   (currently amended) The <u>computer-implemented</u> method of claim 1 wherein dividing said pool of integers into groups includes establishing a hash table that includes hash table pointers, wherein said hash table pointers identify locations in said computer memory of doubly linked lists that are related to said groups of integers.

5.   (currently amended) The <u>computer-implemented</u> method of claim 4 wherein said hash table pointers are related to hash table index values.

6.   (currently amended) The <u>computer-implemented</u> method of claim 4 further including allocating a next free integer by:

using said hash table pointers to search through active doubly linked lists for a next free integer;

if a next free integer exists on an active doubly linked list, then removing the linked list element related to said next free integer from the respective doubly linked list; and

if a next free integer does not exist in any of the active doubly linked lists, then initializing, in said computer memory, a new doubly linked list and removing a linked list element from said new doubly linked list.

7.   (currently amended) The <u>computer-implemented</u> method of claim 6 wherein removing a linked list element includes accessing a free list head element to identify the next free integer in a doubly linked list.

8.   (currently amended) The <u>computer-implemented</u> method of claim 4 further including allocating a specific free integer by:

determining a hash table index related to said specific free integer;

using said hash table index to identify a related hash table pointer;

using said hash table pointer to determine if the related doubly linked list is active;

if said doubly linked list is active, then indexing into the linked list element that is related to said specific integer and removing said linked list element from said doubly linked list;

if said doubly linked list is not active, then initializing a doubly linked list that is related to the group of integers that includes said specific integer, indexing into said doubly linked list to the linked list element that is related to said specific integer, and removing said linked list element, that corresponds to said specific integer, from said doubly linked list.

9. (currently amended) The computer-implemented method of claim 8 wherein removing a linked list element includes:

using the next pointer, of said linked list element that corresponds to said specific integer, to identify the next linked list element in said doubly linked list;

adjusting the previous pointer of said identified next linked list element to exclude, from said doubly linked list, said linked list element that corresponds to said specific integer;

using the previous pointer, of said linked list element that corresponds to said specific integer, to identify the previous linked list element in said doubly linked list; and

adjusting the next pointer of said identified previous linked list element to exclude, from said doubly linked list, said linked list element that corresponds to said specific integer.

10. (currently amended) The computer-implemented method of claim 4 further including freeing a specific integer by:

determining a hash table index related to said specific integer;

using said hash table index to locate the linked list element that is related to said specific integer; and

attaching said linked list element to the doubly linked list of free integers that is related to the group of integers that includes said specific integer.

11.   (currently amended) The <u>computer-implemented</u> method of claim 10 further including:

determining if all integers in said group of integers that includes said specific integer are free; and

if all of the integers in said group of integers that includes said specific integer are free, then deleting the related doubly linked list from said computer memory.

12.   (currently amended) A <u>computer-implemented</u> method for managing a re-usable resource comprising:

dividing a pool of integers into groups that include unique sets of integers; and

initializing, in computer memory, one of said groups of integers as a doubly linked list of free integers in response to a request for a next free integer or a specific free integer if said next free integer or said specific free integer is determined to be in a group of integers that has not been initialized.

13.   (currently amended) The <u>computer-implemented</u> method of claim 12 further including deleting, from said computer memory, an active doubly linked list that represents one of said groups of integers when all of the integers in said one group are free.

14.   (currently amended) The <u>computer-implemented</u> method of claim 12 wherein initializing a doubly linked list includes:

establishing an array of linked list elements that defines said group of integers, wherein each linked list element includes a next pointer for identifying a next linked list element and a previous pointer for identifying a previous linked list element;

forming a doubly linked list, from said linked list elements, that includes all of the integers in said group of integers; and

establishing a head element having a next pointer for identifying one end of said doubly linked list and a previous pointer for identifying the other end of said doubly linked list.

15. (currently amended) The <u>computer-implemented</u> method of claim 12 wherein dividing said pool of integers into groups includes establishing a hash table that includes hash table pointers, wherein said hash table pointers identify locations in said computer memory of doubly linked lists that are related to said groups of integers.

16. (currently amended) The <u>computer-implemented</u> method of claim 15 wherein said hash table pointers are related to hash table index values.

17. (currently amended) The <u>computer-implemented</u> method of claim 15 further including allocating a next free integer by:

using said hash table pointers to search through active doubly linked lists for a next free integer;

if a next free integer exists on an active doubly linked list, then removing the linked list element related to said next free integer from the respective doubly linked list; and

if a next free integer does not exist in any of the active doubly linked lists, then initializing, in said computer memory, a new doubly linked list and removing a linked list element from said new doubly linked list.

18. (currently amended) The <u>computer-implemented</u> method of claim 17 wherein removing a linked list element includes accessing a free list head element to identify the next free integer in a doubly linked list.

19. (currently amended) The <u>computer-implemented</u> method of claim 15 further including allocating a specific free integer by:

determining a hash table index related to said specific free integer;

using said hash table index to identify a related hash table pointer;

using said hash table pointer to determine if the related doubly linked list is active;

if said doubly linked list is active, then indexing into the linked list element that is related to said specific integer and removing said linked list element from said doubly linked list;

if said doubly linked list is not active, then initializing a doubly linked list that is related to the group of integers that includes said specific integer, indexing into said doubly linked list to the linked list element that is related to said specific integer, and removing said linked list element, that corresponds to said specific integer, from said doubly linked list.

20.   (currently amended) The computer-implemented method of claim 19 wherein removing a linked list element includes:

using the next pointer, of said linked list element that corresponds to said specific integer, to identify the next linked list element in said doubly linked list;

adjusting the previous pointer of said identified next linked list element to exclude, from said doubly linked list, said linked list element that corresponds to said specific integer;

using the previous pointer, of said linked list element that corresponds to said specific integer, to identify the previous linked list element in said doubly linked list; and

adjusting the next pointer of said identified previous linked list element to exclude, from said doubly linked list, said linked list element that corresponds to said specific integer.

21.   (currently amended) The computer-implemented method of claim 15 further including freeing a specific integer by:

determining a hash table index related to said specific integer;

using said hash table index to locate the linked list element that is related to said specific integer; and

attaching said linked list element to the doubly linked list of free integers that is related to the group of integers that includes said specific integer.

22.   (currently amended) The <u>computer-implemented</u> method of claim 21 further including:

determining if all integers in said group of integers that includes said specific integer are free; and

if all of the integers in said group of integers that includes said specific integer are free, then deleting the related doubly linked list from said computer memory.

23.   (currently amended) A <u>computer-implemented</u> system for managing a re-usable resource comprising:

computer memory;

a hash table, stored in said computer memory, that includes a plurality of hash table pointers that are related to unique groups of integers from an integer pool, each of said hash table pointers including a field for identifying a location in said computer memory of a doubly linked list that is related to one of said groups of integers; and

means for initializing, in said computer memory, a doubly linked list that represents one of said groups of integers in response to a request for a free integer, wherein said one group of integers includes said free integer.

24.   (currently amended) The <u>computer-implemented</u> system of claim 23 further including means for deleting, from said computer memory, an active doubly linked list that represents one of said groups of integers when all of the integers in said one group are free.

25.   (currently amended) The <u>computer-implemented</u> system of claim 23 wherein each initialized doubly linked list includes:

an array of linked list elements that defines the respective group of integers, wherein each linked list element includes a next pointer for identifying a next linked list element and a previous pointer for identifying a previous linked list element, said linked list elements being formed into a doubly linked list; and

a head element having a next pointer for identifying one end of said doubly linked list and a previous pointer for identifying the other end of said doubly linked list.

26.	(currently amended) The <u>computer-implemented</u> system of claim 23 wherein said hash table pointers are associated with unique hash table index values.

27.	(currently amended) The <u>computer-implemented</u> system of claim 23 further including means for allocating a next free integer by:

    using said hash table pointers to search through active doubly linked lists for a next free integer;

    if a next free integer exists on an active doubly linked list, then removing the linked list element related to said next free integer from the respective doubly linked list; and

    if a next free integer does not exist in any of the active doubly linked lists, then initializing, in said computer memory, a new doubly linked list and removing a linked list element from said new doubly linked list.

28.	(currently amended) The <u>computer-implemented</u> system of claim 27 wherein removing a linked list element includes accessing a free list head element to identify the next free integer in a doubly linked list.

29.	(currently amended) The <u>computer-implemented</u> system of claim 23 further including means for allocating a specific free integer by:

    determining a hash table index related to said specific free integer;

    using said hash table index to identify a related hash table pointer;

    using said hash table pointer to determine if the related doubly linked list is active;

    if said doubly linked list is active, then indexing into the linked list element that is related to said specific integer and removing said linked list element from said doubly linked list;

    if said doubly linked list is not active, then initializing a doubly linked list that is related to the group of integers that includes said specific integer, indexing into said doubly linked list to the linked list element that is related to said specific integer, and

removing said linked list element, that corresponds to said specific integer, from said doubly linked list.

30. (currently amended) The computer-implemented system of claim 29 wherein removing a linked list element includes:

using the next pointer, of said linked list element that corresponds to said specific integer, to identify the next linked list element in said doubly linked list;

adjusting the previous pointer of said identified next linked list element to exclude, from said doubly linked list, said linked list element that corresponds to said specific integer;

using the previous pointer, of said linked list element that corresponds to said specific integer, to identify the previous linked list element in said doubly linked list; and

adjusting the next pointer of said identified previous linked list element to exclude, from said doubly linked list, said linked list element that corresponds to said specific integer.

31. (currently amended) The computer-implemented system of claim 23 further including means for freeing a specific integer by:

determining a hash table index related to said specific integer;

using said hash table index to locate the linked list element that is related to said specific integer; and

attaching said linked list element to the doubly linked list of free integers that is related to the group of integers that includes said specific integer.

32. (currently amended) The computer-implemented system of claim 31 further including means for:

determining if all integers in said group of integers that includes said specific integer are free; and

if all of the integers in said group of integers that includes said specific integer are free, then deleting the related doubly linked list from said computer memory.

33.   (currently amended) A computer-<u>readable medium</u> ~~program product~~ for managing a re-usable resource, said computer-<u>readable medium</u> ~~program product~~ comprising:

computer code for:

dividing a pool of integers into groups that include unique sets of integers; and

initializing, in computer memory, a doubly linked list that represents one of said groups of integers in response to a request for a free integer, wherein said one group of integers includes said free integer.

34.   (currently amended) The computer-<u>readable medium</u> ~~program product~~ of claim 33 further including computer code for deleting, from said computer memory, an active doubly linked list that represents one of said groups of integers when all of the integers in said one group are free.

35.   (currently amended) The computer-<u>readable medium</u> ~~program product~~ of claim 33 wherein initializing a doubly linked list includes:

establishing an array of linked list elements that defines said group of integers, wherein each linked list element includes a next pointer for identifying a next linked list element and a previous pointer for identifying a previous linked list element;

forming a doubly linked list, from said linked list elements, that includes all of the integers in said group of integers; and

establishing a head element having a next pointer for identifying one end of said doubly linked list and a previous pointer for identifying the other end of said doubly linked list.

36.   (currently amended) The computer-<u>readable medium</u> ~~program product~~ of claim 33 wherein dividing said pool of integers into groups includes establishing a hash table that includes hash table pointers, wherein said hash table pointers identify locations in said computer memory of doubly linked lists that are related to said groups of integers.

37.   (currently amended) The computer-readable medium -program product-of claim 36 wherein said hash table pointers are related to hash table index values.

38.   (currently amended) The computer program-readable medium -product-of claim 36 further including computer code for allocating a next free integer by:

using said hash table pointers to search through active doubly linked lists for a next free integer;

if a next free integer exists on an active doubly linked list, then removing the linked list element related to said next free integer from the respective doubly linked list; and

if a next free integer does not exist in any of the active doubly linked lists, then initializing, in said computer memory, a new doubly linked list and removing a linked list element from said new doubly linked list.

39.   (currently amended) The computer-readable medium -program product-of claim 38 wherein removing a linked list element includes accessing a free list head element to identify the next free integer in a doubly linked list.

40.   (currently amended) The computer-readable medium -program product-of claim 36 further including computer code for allocating a specific free integer by:

determining a hash table index related to said specific free integer;

using said hash table index to identify a related hash table pointer;

using said hash table pointer to determine if the related doubly linked list is active;

if said doubly linked list is active, then indexing into the linked list element that is related to said specific integer and removing said linked list element from said doubly linked list;

if said doubly linked list is not active, then initializing a doubly linked list that is related to the group of integers that includes said specific integer, indexing into said doubly linked list to the linked list element that is related to said specific integer, and

removing said linked list element, that corresponds to said specific integer, from said doubly linked list.

41.   (currently amended) The computer-readable medium ~~program product~~ of claim 40 wherein removing a linked list element includes:

using the next pointer, of said linked list element that corresponds to said specific integer, to identify the next linked list element in said doubly linked list;

adjusting the previous pointer of said identified next linked list element to exclude, from said doubly linked list, said linked list element that corresponds to said specific integer;

using the previous pointer, of said linked list element that corresponds to said specific integer, to identify the previous linked list element in said doubly linked list; and

adjusting the next pointer of said identified previous linked list element to exclude, from said doubly linked list, said linked list element that corresponds to said specific integer.

42.   (currently amended) The computer-readable medium ~~program product~~ of claim 36 further including computer code for freeing a specific integer by:

determining a hash table index related to said specific integer;

using said hash table index to locate the linked list element that is related to said specific integer; and

attaching said linked list element to the doubly linked list of free integers that is related to the group of integers that includes said specific integer.

43.   (currently amended) The computer-readable medium ~~program product~~ of claim 42 further including computer code for:

determining if all integers in said group of integers that includes said specific integer are free; and

if all of the integers in said group of integers that includes said specific integer are free, then deleting the related doubly linked list from said computer memory.